Ct Sub 01

one of a result bypass (RRB) operand and an inter-unit result bypass (RIRB) operand, each of which explicitly controls data flow within the pipeline of the processor.

C2

11. (AMENDED) The method of Claim 10 further comprising:

including pipeline stages having instruction decode, writeback and execution stages, and wherein the execution stage has multiple parallel execution units including a current execution unit and an alternate execution unit.

C3

15. (AMENDED) The method of Claim 9 further comprising:

including pipeline stages having instruction decode, writeback and execution stages, and wherein the execution stage has multiple parallel execution units including a current execution unit and an alternate unit; and

sharing source operand data among the parallel execution units of the pipelined processor through the use of a source bypass (RISB) operand in said first register decode value.

C4

28. (AMENDED) A processor comprising:

a first execution unit having at least one first input and a first output;

at least one second execution unit having at least one second input and a second output;

a first input register connected to said at least one first input;

a second input register;

a multiplexer having a first input from said first input register, a second input from said second input register, and an output to said at least one second execution unit; and

a register decode value that specifies bypassing data from said first input register to said at least one second execution unit via said multiplexer.

29. (AMENDED) The processor of claim 28 further comprising:

a first instruction having at least one first source operand and a first destination, said first execution unit processing said first instruction;

a second instruction having at least one second source operand and a second destination operand, said at least one second source operand is the same as said at least one first source operand; and

means for replacing said at least one second source operand with said register decode value.

30. (AMENDED) The processor of claim 29 further comprising:

a register file connected to said first input register and said second input register; and

means for loading said at least one first and said at least one second source operands from said register file.

31. (AMENDED) The processor of claim 29 further comprising:

a memory connected to said first input register; and

means for loading said at least one first and said at least one second source operands from said memory.

32. (AMENDED) The processor of claim 29, said means for replacing further comprising:

an instruction decode mechanism; and

means for said multiplexer choosing input from said first input register.

33. (AMENDED) The processor of claim 29 further comprising:

said register decode value having fewer bits than said at least one second source operand.

34. (AMENDED) The processor of claim 29 further comprising:

a displacement value within said at least one first and said at least one second source operands, said displacement value specifying an effective memory address where data is stored.

35. (AMENDED) The processor of claim 29 further comprising:

a displacement value within said first destination operand, said displacement value specifying an effective memory address where data is stored.

36. (AMENDED) Electromagnetic signals propagating over a computer network comprising:

said electromagnetic signals carrying instruction for execution on a processor for performing the method of claim 9.

37. (AMENDED) A computer readable media comprising:

said computer readable media containing instruction for execution in a processor for performing the method of claim 9.

Please cancel claims 22-27 without prejudice.

Please add new claims 38 et seq. as follows:

38. (NEW) The method of Claim 9 further comprising:

including pipeline stages having instruction decode, writeback and execution stages, and wherein the execution stage has multiple parallel execution units including a current execution unit and an alternate execution unit; and

explicitly controlling data flow within the pipeline stages of the processor through the use of a register result bypass (RIRB) operand to bypass the writeback stage and to allow result data from an alternate execution unit to flow directly to an input execution register.

39. (NEW) The apparatus of Claim 3 wherein the RIRB operand explicitly infers feedback of the data delivered from an alternate one of the execution units to an input register of the current execution unit over a feedback path.

C5
D1

40. (NEW) Apparatus for enabling an instruction to control data flow within a processor of a programmable processing engine, the apparatus comprising:

a pipeline of the processor, the pipeline having a plurality of stages including instruction decode, writeback and execution stages, the execution stage having a plurality of parallel execution units;

a multiplexer connecting parallel execution units; and

an instruction set of the processor, the instruction set defining a register decode value that controls said multiplexer to bypass a source operand from a previous instruction executing in pipeline stages of the processor to the source operand of a current instruction.

41. (NEW) The apparatus of Claim 40 further comprising:

a register file containing a plurality of general-purpose registers for storing intermediate result data processed by the execution units.

42. (NEW) The apparatus of Claim 40 further comprising:

a memory for storing one of transient data unique to a specific process and pointers referencing data structures.

43. (NEW) The apparatus of Claim 40 wherein the register decode value comprises:

a source bypass operand (RISB) that allows source operand data to be shared among the parallel execution units of the pipelined processor.

C5
D1

44. (NEW) The apparatus of Claim 40 wherein the execution units comprise:

a main execution unit and a secondary execution unit, wherein the RISB operand allows the second execution unit to receive data stored at a memory address specified by a displacement operand in the previous instruction executed by the main execution unit.

45. (NEW) The apparatus of Claim 44 wherein the instruction set of the processor comprises:

an opcode directed to the main execution unit, said opcode having sufficient bits to encode a displacement operand;

an opcode directed to the secondary execution unit; and

micro-opcodes to initiate memory prefetches without requiring a dedicated instruction.

46. (NEW) A method for enabling an instruction to control data flow within a pipelined processor of a programmable processing engine, the method comprising the steps of:

defining a register decode value that specifies one of source operand bypassing from a previous instruction executing in pipeline stages of the processor; and

identifying a pipeline stage register for use as a source operand in an instruction containing the register decode value.

47. (NEW) The method of Claim 46 further comprising:

including pipeline stages having instruction decode, writeback and execution stages, and wherein the execution stage has multiple parallel execution units including a current execution unit and an alternate execution unit.

48. (NEW)The method of claim 47 further comprising:

sharing source operand data among the parallel execution units of the pipelined processor through the use of a source bypass (RISB) operand.

49. (NEW) The method of claim 48 further comprising:

receiving data at said alternate execution unit, the data stored at a memory address specified by a displacement operand in a previous instruction executed by said current execution unit of the processor.

50. (NEW) The method of claim 49 further comprising:

realizing two memory references through the use of a single bus operation over a local bus.

51. (NEW) The method of claim 49 further comprising:

encoding the RISB operand with substantially fewer bits that those needed for a displacement address.